Quick Reference Card

LOGICAL SYSTEMS INC.
MILWAUKEE, WI 53223

LDOS Quick Reference
SYNTAX

Upper and lower case may be used interchangeably in LDOS. In the Quick Reference, however, ***italicized*** characters will be used to specify that part of a parameter which may be used as an abbreviation.

Optional Symbology

1. TO, ON, OVER, USING
2. RIGHT CLOSING PARENTHESIS <)>

Abbreviations in Quick Reference :

sw	·	switch : proper response is either ON/YES/Y or OFF/NO/N
{ }	·	encloses default parameters
[ ]	·	encloses optional parameters
< >	·	encloses literal keys
fs	·	filespec
ds	·	devspec
ps	·	partspec
—ps	·	not-partspec
:s	·	source drivespec
:d	·	destination drivespec or only drivespec
p	·	parameter
d	·	decimal number
h	·	hexadecimal number
a	·	alphabetic character
@	·	alphanumeric character
$	·	wildcard character(s)
wd	·	word of the form a@@@@@@@@
|	·	OR (e.g. A | B means A or B)
m1	·	model I
m3	·	model III

LDOS file extension (/a@@) defaults :

/ B A S	·	LBASIC programs
/ C I M	·	DUMP
/ C M D	·	RUN, LOAD, PATCH, and CMDFILE.
/ D C T	·	SYSTEM (DRIVER)
/ D V R	·	SET
/ F I X	·	PATCH files
/ F L T	·	FILTER
/ J B L	·	Joblog files
/ J C L	·	DO, BUILD
/ K S M	·	KSM utility
/ O V d	·	LBASIC overlay (DO NOT USE)
/ S C R	·	Scripsit text file
/ S P L	·	SPOOL command
/ S Y S	·	LDOS SYStem files (DO NOT USE)
/ T X T	·	ASCII p of LScript, LIST, and DUMP

Update Passwords :

(/SYS)	·	.SYSTEM
(/FLT)	·	.GSLTD
(/DVR)	·	.GSLTD
(/CMD)	·	.RRW3
LBASIC	·	.BASIC
(/OVd)	·	.BASIC
CONFIG/SYS	·	.CCC

System Devices :

*KI	·	keyboard
*DO	·	video display
*PR	·	printer output
*JL	·	Job Log (optional)
*SI	·	Standard Input
*SO	·	Standard Output
*CL	·	Comm Line (optional)

LDOS - Alphabetical Quick Reference

APPEND

Attach file or device to end of a file.

APPEND fs to fs [(STRIP)]
APPEND ds to fs [(*E*cho,STRIP)]

ATTRIB

Install, change, remove passwords or access level. Make a file visible or invisible. Lock/unlock non-system visible files. Alter diskette name or Master password.

ATTRIB fs.pw[:d] (*A*CC=wd, *U*PD=wd, *P*ROT=*, [*V*IS *I*NV])
ATTRIB :d (LOCK | UNLOCK, MPW="wd" [,NAME="wd", PW="wd"])

*levels of protection for P=

LEVEL	**Access permitted**
EXEC	execute only
READ	read, execute
WRIT	write, read, execute
NAME	rename, write, read, execute
KILL	all access except ATTRIB
ALL	total access
FULL	same as ALL

AUTO

Modifies power-up sequence by specifying a command to be executed after power-up, reset, or reboot.

AUTO [*] [DOS COMMAND]
* disables bypass and break key

Command may be any LDOS command sequence of up to 31 characters

BACKUP

Duplicates data from a source diskette to a destination diskette.

BACKUP :s to :d [(p,p)]
BACKUP —ps | ps[$] :s to :d [(p,p)]

The parameters are :

*V*IS,*S*YS,*I*NV,*M*OD,*Q*UERY,OLD,NEW,X

*D*ATE="MM/DD/YY—MM/DD/YY"	(from to)
*D*ATE="MM/DD/YY"	(=)
*D*ATE="—MM/DD/YY"	(<=)
*D*ATE="MM/DD/YY—"	(>=)

BOOT

Software reset. Causes diskette in drive Ø to be booted into the system.

BOOT [<CLEAR> | <ENTER> | <D> | <right ARROW>]

Key switches in boot

key	**purpose**
<CLEAR>	no sysgened configuration will occur
<ENTER>	breakable AUTO will be suppressed
<D>	enter debug (no config will occur)
<r ARROW>	m3 only. LDOS video driver will not load.

BUILD

Allows construction of a file consisting of character strings. Using the /KSM extension will prompt for the 26 letters of the alphabet for a KSM data base.

BUILD fs {/JCL} [(HEX,APPEND)]

CLOCK

Turns screen display of "real time" clock on or off.

CLOCK [sw {ON}]

CONV

Allows files to move from a Model III TRSDOS diskette to an LDOS formatted diskette.

CONV ps[$]:s | :s :d [(V | I,S,N | O,Q)]

Where s <> Ø

See BACKUP for parameter words.

COPY

Copies data from one file or device to another file or device.

COPY fs to fs | ps | :d [(*L*RL=n,*C*LONE=sw)]
COPY fs:d (X) . . . single drive copy
COPY ds to fs [(*L*RL=n,*E*CHO)]
COPY ds to ds [(*E*CHO)]
COPY fs to ds

Where 1 <= n <= 256.

CREATE

Creates a filename in directory and optionally reserves record blocks or a specified size.

CREATE fs [(*L*RL=d|256|,*R*EC=d. | SIZE=K)]

Where 1 <= d <= |256| for *L*RL.
Where K = K * 1Ø24 bytes for size.
Size may not be used w/ R or L.

DATE

Sets month, day, and year for directory and LBASIC use. Displays current setting.

DATE [mm/dd/yy]

1 <= mm <= 12, 1 <= dd <= 31, 8Ø <= yy <= 87

DEBUG

Switches LDOS system's debugging utility

debug {sw{ON} [,Ext]}

Debug Command Set

Command function

A	ASCII display
C	single step but perform calls
Dh	starts display at address h
Gh[,h,hl]	exec at h w/ break at h,hl
H	display hexadecimal format
I	single step
Mh	modify memory address h
Rrp h	modify register pair rp to h
S	display full screen mode
U	dynamically update display
X	register display mode
:	advance display
—	decrement display

extended commands

Ba,b,c	move memory block a to b for c bytes
Ea<SPACE>	enter data into memory starting at a
Fa,b,c	fill block a to b with byte c
J	jump over next byte (PC+1)
La,b	locate first occurrence of byte b from address a
N	position cursor to next load block
O	return to LDOS
Pa,b	print memory from a to b
Qa[,b]	display byte at port a or send byte b out port a
Ta	type ASCII starting at a
Va,b,c	compare memory block a to b for c bytes
Wa,b	search for 2 byte b starting at a

Disk read/write utility

a,b,c,d,e,f

Where	a is drive number
	b is cylinder
	c is 1st read/write sector
	d is R to Read, W to Write, * DIR write
	e storage address origin
	f number of sectors

DEVICE

Displays all logical devices, available disk drives. and the state of several system options.

DEVICE [(B=sw{Y},D=sw{Y},S=sw{Y},P=sw{N})]

DIR

Display diskette directory.

DIR [fs[:d] | ps [:d] | —ps[:d] | :d (p,p,etc)]

Parameters are: A,*I*NV,*M*OD,N,S*O*RT=sw,P,*S*YS,*D*ATE= see BACKUP for date codes

DO

Execute a JCL file.

DO [$ | = | *] fs{/JCL} [(@label,p,p);]

DUMP

Writes a block of memory to a disk file. File may be in load module or ASCII formats.

DUMP fs (*S*TART=d | h,*E*ND=d | h,*T*RA=d | h[*A*SCII, ETX=d | h)]

Default extension {/CIM} | {/TXT} - if ASCII
Where h is of the format X'hhhh'.
S address > X'55ØØ'. E address >= S

FORMAT

Writes cylinders, sectors, and directory to a blank diskette so that it may be used by LDOS.

FORMAT [:d (p,p,etc)]

The parameters are :
NAME="wd"{LDOSDISK},MPW="wd"{PASSWORD},
SDEN | DDEN,SIDES=1 | 2,CYL=d(<=96),
STEP=d(Ø-3),*Q*UERY=sw{Y},SYSTEM,ABS

FILTER

Executes a program which modifies the I/O path of a specified device.

FILTER ds using fs [/FLT] [(p,p,etc)]

Parameters depend on the file employed.

FREE

Shows used and available space and files on each diskette or displays a space map of a specific diskette.

FREE [:d (P)]

HITAPE

Permits 15ØØ baud cassette I/O in LBASIC or CMDFILE on a Model III computer.

HITAPE

JOBLOG

Engages the LDOS Joblog device.

set *JL to ***JL**/DVR using fs [ds

KEYBOARD DRIVER

Allows advanced keyboard features.

set *KI to ***KI**/DVR [(***T**YPE,***J**KL,***D**ELAY=d[,***R**ATE=d2)]

Where d1 >= 1Ø(3Ø), and d2 >= 1(3)

KEY STROKE MULTIPLY

Allows use of files containing phrases to be used as direct keyboard inputs.

FILTER *KI to ***KSM**/FLT using fs[/KSM] [(***E**NTER=d)]

Where 1 <= d <= 255 (59)

KILL

Deletes a specified device or file from the system.

KILL fs [ds

LCOMM

Communications program

LCOMM ds [(XS=X'ffff',XR=X'ffff',NULL=sw(Y))]

Where ff is hexadecimal character to translate from and tt is character to translate to.

LIB

Displays LDOS command libraries.

LIB

LINK

Joins two devices so that output directed to the first device is also sent to the second device. Input requested by the first device can be supplied by the second device.

LINK ds to ds

LIST

Sends a listing of a file to the video display or line printer.

LIST fs [(***A**SCII**8**,***N**UM,***H**EX,***T**AB,P,LINE=d,***R**EC=d, ***L**RL=d)]

LOAD

Reads a load module format file into memory but does not execute it.

LOAD [(X)] fs

LOG

Update drive control table after switching diskettes.

LOG :d

MEMORY

Reserves a portion of memory, displays current HIGH$, modifies a memory address, fills RAM with X'ØØ' from X'52ØØ' to HIGH$, or jumps to a specified memory location.

MEMORY [(*H*IGH=h,*A*DD=h,*W*ORD=h,*B*YTE=h,
*G*O=h)]
MEMORY[(*H*IGH=d,*A*DD=d,*W*ORD=d,*B*YTE=d,
*G*O=d)]
MEMORY (*C*lear)

Where h is of the form X'hhhh' or X'hh' (byte), W= cannot be X'ØØØØ' or Ø, and B= cannot be X'FF' or 255

MINIDOS

Accesses certain LDOS functions without having to be at the LDOS Ready prompt.

FILTER *KI using ***MINIDOS***/FLT

After the filter is operational, press the <CLEAR> <SHIFT> and one of the following keys all together

Key	function
<C>	switch the CLOCK display on or off
<D>	enter DEBUG if active
<F>	display FREE space for drive 0
<K>	kill a file
<P>	send a character to the printer
<Q>[d/ext]	display a directory
<R>	repeat the last LDOS command
<T>	issue a top of form to printer

PATCH

Used to make minor changes or repairs to existing program or data files.

PATCH fs[/CMD] using fs[/FIX] [(YANK)]
PATCH fs[/CMD] using (information in patch format)

PDUBL

Executes a driver which enables some non Radio Shack double density boards on the Model I.

PDUBL

PR

Formats data sent to printer

FILTER *PR using ***PR***/FLT [(p,p, . . .)]

The parameters are: *A*DDLF,*C*HARS=dd,*F*F,*I*NCRL,
*I*NDENT=d[Ø],*L*INES=d[P parm],*M*ARGIN=d[Ø],
*P*AGE=d[66],PORT=d,*S*LINE=Ø | 1,*T*AB,
*X*LATE=X'ffft'

Where Ø <= d <= 255 and ff is a hex character to be converted to hex tt.

PURGE

Allows controlled multiple file deletions.

PURGE :d [(*Q*UERY=sw{ON},MPW="wd",*I*NV,*S*YS, *D*ATE=" ")]

PURGE—ps | [$] :d [(p,p as above)]

Where D= is the same parlance as in BACKUP.

RDUBL

Executes a driver which enables the Radio Shack double density board on the Model I.

RDUBL

RENAME

Changes the name of a file.

RENAME fs to fs | ps

REPAIR

Updates and corrects information on certain types of diskettes to make them usable by LDOS.

REPAIR :d (ALIEN)

RESET

Cancel specified or all links, routes, sets, drivers, or filters.

RESET [ds]

ROUTE

Re-directs I/O for a specific device to another device or to a bit bucket.

ROUTE ds to (NIL) | fs | ds

RUN

Loads and executes a machine language program (load file format).

RUN [(X)] fs [(p,p depends on fs)]

SET

Engages a driver program to enable a device.

SET ds to fs{/DVR} [(p,p depends on fs)]

SPOOL

Establishes a FIFO buffer for a specified device (usually a printer).

SPOOL ds (OFF) | to fs{/SPL} (*M*EM=K{1},*D*ISK=K{Ø}

Where K is 1Ø24 bytes.

SYSTEM

Configures user definable areas of the LDOS system.

SYSTEM (p,p,etc)

The parameters are :

parm	function
ALIVE=sw{ON}	displays activity character
BASIC2	activates ROM BASIC
BLINK=sw | d	toggles blinking cursor and selects cursor character where d is an ASCII number
BLINK,LARGE|SMALL	large toggles char 143 small toggles char 136
BREAK=sw{ON}	enable/disable break key

BSTEP=d	(Ø-3) bootstrap step rate
DATE=sw{ON}	toggle date prompt
DRIVE=d,p,p . . .	sets disk drive parms: CYL=d; 35 <= d <= 96 DELAY=sw{ON}: toggles 5" time before 1st read DISABLE: removes drive from DCT ENABLE: places drive in DCT STEP=d: Ø <= d <= 3 DRIVER="fs"|/DCT}: establishes disk driver WP=sw{ON} :toggles write-protect
FAST	enables high speed mod
GRAPHIC	appraises system of printer graphic character set
SLOW	disables high speed mod
SVC	load SVC table in high memory
SYSGEN=sw{ON}	alters, creates or removes user configuration.
SYSRES=d	reside system overlays 1 <= d <= 12 AND d <> 6 |7
SYSTEM=d	assigns system drive
TIME=sw{ON}	toggles time prompt
TYPE=sw{ON}	toggles type ahead task
UPDATE=sw{ON}	toggles Model I date to change at 23:59:59

TIME

Sets or displays "real time" clock.

TIME [hh:mm:ss]
Where : ØØ <= hh <= 23 & ØØ <= mm \| ss <= 59

TRACE

Displays PC register in upper right corner of video.

TRACE=sw{ON}

TWOSIDE

Allows use of two sided SDEN disk drives with the Model I.

TWOSIDE

VERIFY

Forces all disk writes to be verified with a read after write operation.

VERIFY=sw{ON}

LCOMM Programmed Function Keys

***KI — <CLR><1>**
Toggles keyboard except for the PF keys.

***DO — <CLR><2>**
Toggles video display on and off.

***PR — <CLR><3>**
Toggles printer device on and off.

***CL — <CLR><4>**
Toggles communications.

***FS — <CLR><5>**
Engages file send.

***FR — <CLR><6>**
Engages file receive.

DTD — <CLR><7>
Dump To Disk. Used to write a memory buffer obtained by *FR to disk.

MENU — <CLR><8>
Prints menu/status display to video.

ID — <CLR><9>
Opens a file for transmit or receive. Modified by *FR or *FS.

RESET — <CLR><Ø>
Close receive or send file. Clear a device buffer.

ON — <CLR><:>
Used to toggle devices on or engage transmission or reception of a file.

OFF — <CLR><—>
Used to toggle devices off or halt transmission or reception of a file.

DUPLEX — <CLR><SH><1>
Toggles duplex on (half-duplex) and off (full-duplex).

ECHO — <CLR><SH><2>
Toggles host retransmit of received data (full-duplex for terminal).

ECHO-LINEFEED — <CLR><SH><3>
Toggles linefeed character after CR.

ACCEPT-LINEFEED — <CLR><SH><4>
Toggles selective linefeed mode.

REWIND — <CLR><SH><5>
Rewinds files opened by *FS or *FR.

PEOF — <CLR><SH><6>
Positions receive file to end for append.

DCC — <CLR><SH><7>
Displays control characters received.

CLS — <CLR><SH><8>
Clear Local Screen. No effect on buffers.

8-BIT — <CLR><SH><9>
Toggle eight bit mode.

HANDSHAKE — <CLR><SH><*>
Engage X-On/Off communications protocol.

EXIT — <CLR><SH><=>
Return to LDOS Command Level.

LBASIC Quick Reference
SYNTAX

Abbreviations in Quick Reference :

sw	· switch : proper response is either ON/YES/Y or OFF/NO/N
{ }	- encloses default parameters
[ ]	· encloses optional parameters
< >	· encloses literal keys
fs	· filespec
d	· decimal number
db	· decimal number range 1 <= x <= 15
d%	· decimal number -32768 <= x <= 32767
hb	- hexadecimal number prefixed by &H
|	· OR (e.g. A | B means A or B
m3	· model III
var	· any variable
var*	· any numeric variable
var%	· an integer variable
var!	- a single precision variable
var#	· a double precision variable
var$	· a string variable

LBASIC Terminology

Command · A keyword and its parameters which is normally executed at the LBASIC "Ready" prompt and is not usually found in a program line.

Function · A keyword and its parameters that must pass information to either an output statement or a variable but cannot stand alone on a program line.

Statement - A keyword which performs a job usually while a program is executing.

Entering LBASIC

To enter LBASIC at LDOS Ready type:

LBASIC [* | (p,p . . .) command(s)]

The parameters are :

parm	function
*B*LK=sw{ON}	toggles block file mode
*F*ILES=db{3}	max number of files (can be Ø)
*M*EM=d | hb	highest memory address to be used by LBASIC.
*E*XT=sw{ON}	toggles default file extension
{*H*IGH} | *L*OW	m3 toggles 15ØØ/5ØØ baud cassette
command(s)	any BASIC command, will execute upon entry to LBASIC
*	used to re-enter LBASIC with program and variables intact.

LBASIC Abbreviated Commands

key	function
A	same as AUTO
D	same as DELETE
E	same as EDIT
L	same as LIST
.	same as LIST.<ENTER>
,	same as EDIT. <ENTER>
<up arrow>	list next lower line
<down arrow>	list next higher line
<left arrow>	list first program line
<right arrow>	list last program line
<shift><@><space>	single step paused program

LBASIC Alphabetical Quick Reference

& H
Prefix for hexadecimal constant.

var*=&HFFØØ

& O
Prefix for an octal constant

var*=&O11

CLOSE
Disables read/write to a specified or all OPEN disk files.

CLOSE [db, db . . .]

Where db is the file buffer to disable.

CMD (general)
Performs certain LDOS library and utility commands without leaving LBASIC.

CMD "dos command"

Where dos command is any library or utility that does not utilize high memory .

CMD "*"
Sends the screen contents to a printer.

CMD "A"
Returns to LDOS and aborts any active DO file.

CMD "B", "sw"
Enable/disable <BREAK> key.

CMD "D"
Turns on and enters system debugger.

CMD "D", "sw"
Turns on but does not enter debugger until <BREAK> key is pressed or <CLEAR><SHIFT><D> is pressed with MINIDOS active.

CMD "E"
Returns last LDOS error.

CMD "I", "dos command"
Executes dos command and then exits LBASIC.

CMD "L", "fs"

Reads a load module format file (machine language) into memory.

CMD "N"

Renumbers LBASIC line references and numbers.

CMD "N [! a,b,c,d]"

Where ! stops scan for errors prior to renumbering, a is the current line number from which to begin (1), b is the new line number for a (2Ø), c is the increment between line numbers (2Ø), and d is the last line to be renumbered (65529).

CMD "O"

Sorts a single dimension string array.

CMD"O",a,b

Where a is the number of elements to be sorted and b is the array variable (var$(+d%)) to begin.

CMD "P", variable

Returns printer status to specified variable.

CMD "R"

In Model I, enables the interrupts.
In Model III, turns on clock display.

CMD "S"

Normal exit from LBASIC.

CMD "T"

In Model I, disables the interrupts.
In Model III, turns off clock display.

CMD "X"

Produces a cross reference of variables and line numbers for currently loaded program.

CMD"X [ds | fs p,t]"

Where t is an optional title to be printed at the top of each page, ds | fs is the destination device for the listing (*DO).

The optional parameters are :

—V for all variables
—var only the specified variable
—L all line numbers
—d only the specified line number
—none all variables and line numbers

CVD

Function to convert an 8 byte string to a double precision decimal number. (See MKD$)

```
var#=CVD(var$)                    PRINT CVD(var$)
```

CVI

Function to convert a 2 byte string to an integer precision decimal number. (See MKI$)

```
var%=CVI(var$) : IF CVI(var$) < 2ØØØØ THEN 2ØØ
```

CVS

Function to convert a 4 byte string to a single precision decimal number. (See MKS$)

```
var!=CVS(var$) : PRINT USING"### ##";CVS(var$)
```

DEF FN

Defines a user function.

DEF FNnam(var,var. . . .)=exp.

Where nam is the function name (same as a variable name) which must match the type of the returning value, var is a variable to which the function output will be written, and expression is how the variables are to be manipulated.

DEF FNP!(var1,var2)=SQR(var1*var1+var2*var2)

DEFUSR

Defines entry point of a machine language subroutine.

DEFUSRd=d% | hb

Where Ø <= d <= 9 and d% | hb is the entry address of the machine language subroutine.

EOF

Function which determines if the end of an open file has been reached.

var*=EOF(db), IF EOF(db) THEN 1ØØØ, PRINT EOF(db)

The function returns a Ø for false and a -1 for true.

FIELD

Partitions buffer of an open random access file.

FIELDdb, d as var1, d as var2 etc.

Where d is the number of characters that each variable is to occupy. Total chars must be equal or less than the number of bytes declared when the file was opened.

FN

Calls a user defined function

var=FNnam(var,var...)

Where nam is a previously defined function (DEF FNnam). From the example in DEF FN:

var=FNP!(3,4) would return a value of 5 in var.

GET

Retrieve a record from a random file.

GETdb[,d%]

If d% is not used, the command will retrieve the current record +1.

INPUT#

Input data from a sequential file. (see PRINT#)

INPUT#db,var,var,...

The variables should be of the same type as were written to the file.

INSTR

Function which locates a position of a string contained within a string.

INSTR([d,]var1$,var2$)

Where 1 <= d <= 255, and var1$ is searched starting at the d character {1}.

IF INSTR(var1$,var2$) THEN etc.

P=INSTR(var1$,var2$)

ON INSTR(var1$,var2$) GOTO etc.

KILL

Remove a disk file from a diskette.

KILL"fs" |var$

LINEINPUT

Assigns a string of keyboard characters to a variable. Similar to INPUT but inputs all characters until <ENTER> is pressed.

LINEINPUT var$

LINEINPUT#

Input a line from a disk file until either encountering a carriage return or 255 characters.

LINEINPUT#db,var$

LOAD

Command to retrieve a BASIC program from the disk. It is possible to run the file with an optional "R".

LOAD"fs"[/BAS] |var$ [,R]

LOC

Function which returns the current record number of an open file.

var%=LOC(db) PRINT LOC(db)

LOF

Function which returns the last record number of an open file.

var%=LOF(db) PUTdb,LOF(db)+1

LSET

Statement which passes information from a variable to a random file buffer by setting a FIELDed string variable from the left and padding the right with spaces if necessary.

LSETvar$=MKI$(var%) LSETvar$=var2$

MERGE

Command which allows integration of a disk file program to the current program in memory. File to MERGE must be in ASCII format.

MERGE"fs"[/BAS] |var$

MID$

Retains its usage but also now has the capability of being on the left side of an equation. In this capacity will replace a portion of a string.

MID$(var$,a,b)=var2$ |"string"

Alters var$ starting at position a, 1 <= a <= LEN(var$), for b positions, 1 <= b <= LEN (var$) AND <= LEN(var2$), by substituting the contents of var2$.

MKD$

Function which changes a double precision number to an 8 byte string. (see CVD)

var$=MKD$(var#) LSET var$=MKD$(var#)

MKI$

Function which changes an integer precision number to a 2 byte string. (see CVI)

var$=MKI$(var%) RSET var$=MKI$(var%)

MKS$

Function which changes a single precision number to a 4 byte string. (see CVS)

```
var$=MKS$(var!)      RSET var$=MKS$(var!)
```

OPEN

Prepare to access a disk file. Sequential Access:

```
OPEN"O | E[O | N]",db,"fs" | var$
OPEN"I",db"fs" | var$
```

mode	function
I	existing file for input
O	for output
OO	existing file for output
ON	new file for output
E	for output and extend
EO	extend existing file for output
EN	new file for output

Random Access:

```
OPEN"R[O | N]",db,"fs" | var$[,LRL]
```

mode	function
R	open random file
RN	open new random file only
RO	open existing file only

Where ∅ <= LRL <= 255 {∅} (∅ means 256) is the logical record length of the buffer db.

PRINT#

Allows the writing of data to a sequential access file.

```
PRINT#db,var%;var2%;var!;var2!;var#;var2# . . .etc
PRINT#db,var$;",";var2$;",";var3$ . . .etc
PRINT#db,var%;var$;",";var!;var$;",";var2%;var3%
```

PRINT# USING

Output data to a sequential file using a specified format.

```
PRINT#db USING"format" | var$;var
```

PUT

Write a record to an open random file.

```
PUTdb[,d%]
```

If d% is not used current record +1 is written

RESTORE

Augments the old RESTORE command by allowing the line number specification to which the data pointer is to be set.

```
RESTORE ln
```

Where ln = a BASIC line number. When used in this fashion, this statement must be first on a program line.

RSET

Statement which passes information from a variable to a random file buffer by setting a FIELDed string variable from the right and padding the left with spaces if necessary.

```
RSETvar$=MKI$(var%)      RSETvar$=var2$
```

RUN

When used with a filespec, will load and execute an LBASIC program stored on a diskette.

RUN"fs"[/BAS] | var$ [,R | V,ln]

Where R would signal that all currently open files remain open and V would additionally preserve all variable values. ln is an optional parameter which is the line number at which to start execution.

SAVE

Saves the current program in memory to disk.

SAVE"fs"[/BAS] | var$ [,A]

Using the ,A parameter saves the file in an ASCII format.

SET EOF

Reset the end of file marker in an open random file in order to "shrink" the file size. After SET EOF is performed the last accessible record will now be the current LOC of the file. Usually done by performing a GET to the last desired record.

SET EOF db

TIME$

Function which returns a 17 byte string of the current setting of the real time clock and date.

var$=TIME$ PRINT LEFT$(TIME$,8)

The format of the string is :
mm/dd/yy hh:mm:ss

USR

Function which causes execution of a machine language subroutine previously defined in a DEF USR statement.

var*=USRn(var% | d%)

Where var is a numeric variable which could contain a value passed from the subroutine, $\emptyset <= n <= 9$, and the value passed to the subroutine is an integer or integer variable.

LBASIC Error Number Reference

Alphabetical Order

message	ERR number
Bad File Mode	1Ø8
Bad File Name	128
Bad File Number	1Ø4
Bad Record Number	126
Blocked File Error	14Ø
Direct Statement in File	132
Disk Full	122
Disk I/O Error	114
Disk Write Protected	136
Field Overflow	1ØØ
File Access Denied	138
File Already Exists	116
File Already Open	11Ø
File Not Found	1Ø6
Input Past End	124
Internal Error	1Ø2
Protection cleared Memory	144
System Command Aborted	142
Too Many Files	134

Numerical Order

message	ERR number
Field Overflow	1ØØ
Internal Error	1Ø2
Bad File Number	1Ø4
File Not Found	1Ø6
Bad File Mode	1Ø8
File Already Open	11Ø
Disk I/O Error	114
File Already Exists	116
Disk Full	122
Input Past End	124
Bad Record Number	126
Bad File Name	128
Direct Statement in File	132
Too Many Files	134
Disk Write Protected	136
File Access Denied	138
Blocked File Error	14Ø
System Command Aborted	142
Protection cleared Memory	144

ASCII CHARACTER SET

LSD \ MSD	0	1	2	3	4	5	6	7
	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*		J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	US	/	?	O	-	o	DEL

HEX AND DECIMAL CONVERSION

LSD →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	2
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	3
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	4
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	5
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	6
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	7
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	8
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	9
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	A
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	B
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	C
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	D
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	E
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

NIBBLE CHART

Decimal	Hexadecimal	Binary
Ø	Ø	ØØØØ
1	1	ØØØ1
2	2	ØØ1Ø
3	3	ØØ11
4	4	Ø1ØØ
5	5	Ø1Ø1
6	6	Ø11Ø
7	7	Ø111
8	8	1ØØØ
9	9	1ØØ1
1Ø	A	1Ø1Ø
11	B	1Ø11
12	C	11ØØ
13	D	11Ø1
14	E	111Ø
15	F	1111

LDOS Error Messages

Error #	Error Message
ØØ	No Error
Ø1	Parity error during header read X'Ø1'
Ø2	Seek error during read X'Ø2'
Ø3	Lost data during read X'Ø3'
Ø4	Parity error during read X'Ø4'
Ø5	Data record not found during read X'Ø5'
Ø6	Attempted to read system data record X'Ø6'
Ø7	Attempted to read locked/deleted data record X'Ø7'
Ø8	Device not available X'Ø8'
Ø9	Parity error during header write X'Ø9'
1Ø	Seek error during write X'ØA'
11	Lost data during write X'ØB'
12	Parity error during write X'ØC'
13	Data record not found during write X'ØD'
14	Write fault on disk drive X'ØE'
15	Write protected disk X'ØF'
16	Illegal logical file number X'1Ø'
17	Directory read error X'11'
18	Directory write error X'12'
19	Illegal filename X'13'
2Ø	GAT read error X'14'
21	GAT write error X'15'
22	HIT read error X'16'
23	HIT write error X'17'
24	File not in directory X'18'
25	File access denied X'19'
26	Full or write protected disk X'1A'
27	Disk space full X'1B'
28	End of file encountered X'1C'
29	Record number out of range X'1D'
3Ø	Directory full - can't extend file X'1E'
31	Program not found X'1F'
32	Illegal drive number X'2Ø'
33	No device space available X'21'
34	Load file format error X'22'
35	Memory fault X'23'
36	Attempted to load read only memory X'24'
37	Illegal access attempted to protected file X'25'
38	File not open X'26'
39	Device in use X'27'
4Ø	Protected system device X'28'
>4Ø	Unknown Error Code

LOGICAL® SYSTEMS INC.

MILWAUKEE. WI 53223